

# A Minimum Cost Network Flow Model for the Maximum Covering and Patrol Routing Problem

R. Dewil<sup>1</sup>, P. Vansteenwegen, D. Cattrysse, D. Van Oudheusden

*KU Leuven, Centre for Industrial Management, Traffic and Infrastructure. Celestijnenlaan 300A, 3001 Heverlee, Belgium*

---

## Abstract

This paper shows how the maximum covering and patrol routing problem (MCPRP) can be modeled as a minimum cost network flow problem (MCNFP). Based on the MCNFP model, all available benchmark instances of the MCPRP can be solved to optimality in less than 0.4s per instance. It is furthermore shown that several practical additions to the MCPRP, such as different start and end locations of patrol cars and overlapping shift durations can be modeled by a multi-commodity minimum cost network flow model and solved to optimality in acceptable computational times given the sizes of practical instances.

**Keywords:** Routing, Problem structuring, Minimum Cost Network Flow Problem, Multi-Commodity

---

## 1. Introduction

The maximum covering and patrol routing problem (MCPRP) was introduced by Keskin et al. [12] and is used to assist traffic enforcement. A typical method for state troopers is to patrol "hot spots" which are certain locations on highways where particular types of crashes (e.g. crashes caused by speed or driving under influence) frequently occur [20, 2]. Furthermore, these hot spots are only active during certain time windows. Due to limited resources, not all active hot spots can be patrolled. Therefore, an optimization problem to route patrol cars in a way that maximizes hot spot coverage appears to be appropriate. Keskin et al. [12] model the MCPRP as a variant of the orienteering problem [23, 24], prove that their model is NP-hard, and present two heuristics, a local search heuristic and a tabu search heuristic which determine good quality solutions in short periods of time. The authors claim that a heuristic solution instead of an exact technique is preferred for their model since it is important for the practitioner to obtain a good solution quickly.

Having an efficient and effective solution method to solve the MCPRP is useful since the problem often

appears as a sub problem in larger problems. Li and Keskin [14] consider a bi-objective multi-period patrol routing problem. The multi-period aspect appears through the introduction of intermediate temporary stations in the patrol routes. Li and Keskin develop a heuristic that exploits the hierarchical structure of the problem by decomposing the problem in a location and a routing problem. An effective solution method for the MCPRP can be incorporated in such a framework to solve the routing problem more efficiently. If an orienteering problem approach is taken, this is similar to the orienteering problem with hotel selection (OPHS) of Divsalar et al. [6, 7] which considers a multi-period tourist trip planner application. Very recently, Çapar, Keskin, and Rubin[3] reconsidered the MIP formulation for the MCPRP and also used a set of domination rules to greatly simplify the MIP formulation. The new MIP formulation is able to solve their set of benchmark instances to optimality within reasonable calculation times. It also allows several extensions to be tested, such as, letting troopers start from their homes, allowing delayed starts and intra-day diversion.

Other patrol routing problems that require rerouting when incidents in the network occur during the execution of a patrol scheme are considered by Moonen et al. [17], Takamiya and Watanabe [21], Chen [4], and Portugal and Rocha [19]. These problems can pos-

---

<sup>1</sup>Corresponding author. Tel.: +32 16 322566

Email address: reginald.dewil@kuleuven.be

sibly also benefit from a fast solution method for rerouting patrol cars.

Modeling the MCPRP as a minimum cost network flow problem (MCNFP) [26], which is known to be solvable in polynomial time [18] would be a major improvement over existing models in the literature. This is an example of exploiting the network structure of a problem to gain computational efficiency and it underscores the importance of selecting an appropriate model for a particular problem. In this particular case, the network structure is uncovered in the following sections step by step. For problems where the network structure is not obvious, a systematic approach such as the *netform concept* described by Glover et al. [10] can be used to explicit this structure.

Several other patrol routing applications, each with their own specific constraints can be found in the literature. Some of the more recent ones are Lou et al. [15], Willemse et al. [25], and Chircop et al. [5]. These applications differ substantially from the basic MCPRP as described in [12] which prohibits a straightforward application of the proposed MCNFP reformulation. Because of the binary nature of the "hot spot profits", these applications share more characteristics with the orienteering problem or the rural postman problem [8] than with the MCNFP.

The remainder of this paper is structured as follows. In section 2, we show how the original MCPRP problem can be modeled as an MCNFP. In section 3, we discuss the results of the computational experiments applied on the instances of Keskin et al. [12]. In section 4, we show how practical extensions to the MCPRP can be modeled by adding a multi-commodity aspect to the MCNFP [22] and discuss the results of some computational experiments on artificial data sets in section 5. Finally, in section 6, we provide our conclusions and recommendations for future research.

## 2. Problem analysis

Given a set of patrol cars and a set of hot spots, the objective of the MCPRP consists of finding a set of routes for the patrol cars that maximizes the time spent in hot spot locations. The travel times between hot spots are constant and known beforehand. Each hot spot is only active during certain time windows. All cars have the same shift start and end times and the same start and

end locations. In addition, multiple cars being present in the same hot spot at the same time does not increase the objective function [12]. A car can enter and leave a hot spot at any given time but only collects "gain" for the duration that a car remains in the hot spot within the hot spot's time window. Figure 1 represents a possible routing of two patrol cars in a graphical way. Both patrol cars are allowed to leave the depot at time 0 and need to return at the end of their shift to the depot. The horizontal dotted lines represent the fact that a car can arrive at a hot spot before the hot spot's time window is open. This time does not provide any *gain* and is called *dead time*. The angled dotted lines represent the actual movement of the cars between hot spots. In the basic MCPRP, when making abstraction of the underlying road network, the travel times between hot spots are assumed to be constant and subject to the triangle inequality. Additionally, the gain per minute can be different for each hot spot but is assumed to be constant for the full duration of the hot spot's time window.

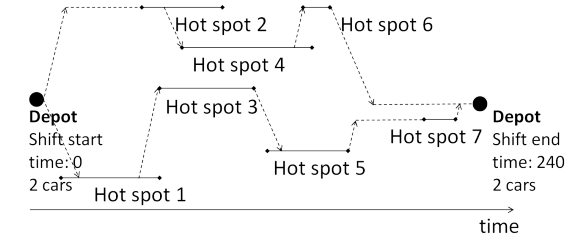


Figure 1: The maximum covering and patrol routing problem with two cars

### 2.1. Domination rule

The transformation of the MCPRP into a minimum cost network flow problem relies on the fact that a lot of solutions of the MCPRP are dominated. In an optimal solution, a patrol car will always stay at its current hot spot until the end of its time window unless another hot spot becomes available earlier and a larger *gain* can be obtained in the latter hot spot, taking the travel time between hot spots into account.

Thus, patrol cars will enter a hot spot  $i$  either at the opening of its time window ( $t_{o,i}$ ) or at the closing time of another hot spot  $j$ 's time window plus the travel time between both hot spots, denoted as  $t_{a,ji}$ . Similarly, a hot spot  $i$  will only be exited at the closing of its time window ( $t_{c,i}$ ) or at the opening of another hot spot  $j$ 's time window minus the travel time between both hot spots, denoted as  $t_{b,ij}$ . This results in splitting each hot spot's

time window  $i$  into segments, referred to as *time sections*. The start and end points of the time sections of a hot spot  $i$  are defined by  $t_{o,i}$ , the  $t_{b,ij}$ 's, the  $t_{a,ji}$ 's and  $t_{c,i}$ . From this point on, for the sake of brevity, we will refer to a time section as a section.

To further clarify the process of identifying sections required to optimize the problem, consider the example shown in figure 2(a). The network consists of three hot spots with overlapping time windows. In order to identify the sections, a forward and a backward pass over all hot spots is executed. Figure 2(b) shows the forward pass. From every hot spot's end time (including the source), the arrival time at any other hot spot (including the sink) is determined, or in other words, all  $t_{a,ji}$ 's are determined. This process is depicted by the dotted lines. The gray lines mean that the end time of the previous hot spot's time window plus the travel time is smaller than the start time of the next hot spot's time window. Likewise, the red dotted lines mean that the possible arrival time in the next hot spot falls outside the hot spot's time window. As a consequence, these "dotted" relationships will not create additional sections. However, the green dotted lines emanating from hot spot 2 will cause splits in the time windows of hot spots 1 and 3. Figure 2(c) shows the backward pass which determines all  $t_{b,ij}$ 's. Ultimately, figure 2(d) shows all identified sections.

To reiterate, we define a section as a time segment of the hot spot's time window. It is characterized by the fact that, in an optimal solution, a car can only enter a section at the start of the section's time window and only leave it at the end of the section's time window. We formulate this domination rule as the following lemma:

**Lemma 1.** Any routing where a car moves from a section  $S$  before the end time of the section to another section  $T$  will be dominated by either the solution where the car leaves at the end time of section  $S$  or the solution where the car did not enter section  $S$  and immediately entered section  $T$ .

A proof of this lemma can be found in Appendix A.

Without loss of generality, the patrol routing problem is then redefined as: "Find a set of routes for the patrol cars that maximizes the time spent (or *gain*) at the sections under the additional constraint that a section can only be visited by at most one car. It should be noted, however, that it is still physically possible for two patrol cars to be present at the same hot spot location at the same time. However, this implies that one of the

cars is waiting to move to another hot spot or arrived too early at the current hot spot. This physical situation will be modeled by one car actually visiting the section, collecting the gain, and one car using a travel arc that physically passes by this section while moving from or to another hot spot, not collecting the gain and thus not visiting the section.

We will now first define the general Minimum Cost Network Flow Problem (MCNFP) and then we will explain how these "sections" can be used to model the Maximum Covering Patrol Routing Problem (MCPRP) as an MCNFP.

## 2.2. Minimum Cost Network Flow Problem

The Minimum Cost Network Flow Problem consists of finding the cheapest possible way of sending a given amount of flow through a network, where a cost and capacity is associated with each arc in the network. It can be modeled by the following linear program, where  $f_{ij}$  represents the flow from node  $i$  to node  $j$ ,  $a_{ij}$  represents the cost of moving one unit of flow over arc  $ij$ ,  $c_{ij}$  represents the capacity of arc  $ij$  and  $D$  represents the required flow.

$$\text{Min} \sum_i \sum_j a_{ij} f_{ij} \quad (1)$$

$$\sum_i f_{ij} = \sum_k f_{jk} \quad \forall j \neq \text{source}, \text{sink} \quad (2)$$

$$\sum_i f_{\text{source},i} = \sum_i f_{i,\text{sink}} = D \quad (3)$$

$$f_{ij} \leq c_{ij} \quad \forall i, j \quad (4)$$

$$f_{ij} \in \mathbb{R}_{\geq 0} \quad \forall i, j \quad (5)$$

Constraint set (2) ensures that an equal amount of flow leaves a given node as enters it. (3) ensures that  $D$  amount of flow leaves the source and enters the sink. (4) ensures that the flow through an arc never exceeds the capacity of the arc and (5) limits the flow variables to the domain of non-negative real numbers  $\mathbb{R}$ . The MCNFP contains the integrality property which states that when all supply and demand values ( $D$ ) and all capacity constraints are integer, all basic solutions are integral [26].

## 2.3. Transforming the MCPRP into the MCNFP

In order to transform the MCPRP into the MCNFP, it should be noted that the graph of the MCNFP will be based on the sections defined above and will be an abstraction of the road network. The nodes represent the start or end of patrolling a certain hot spot during a given

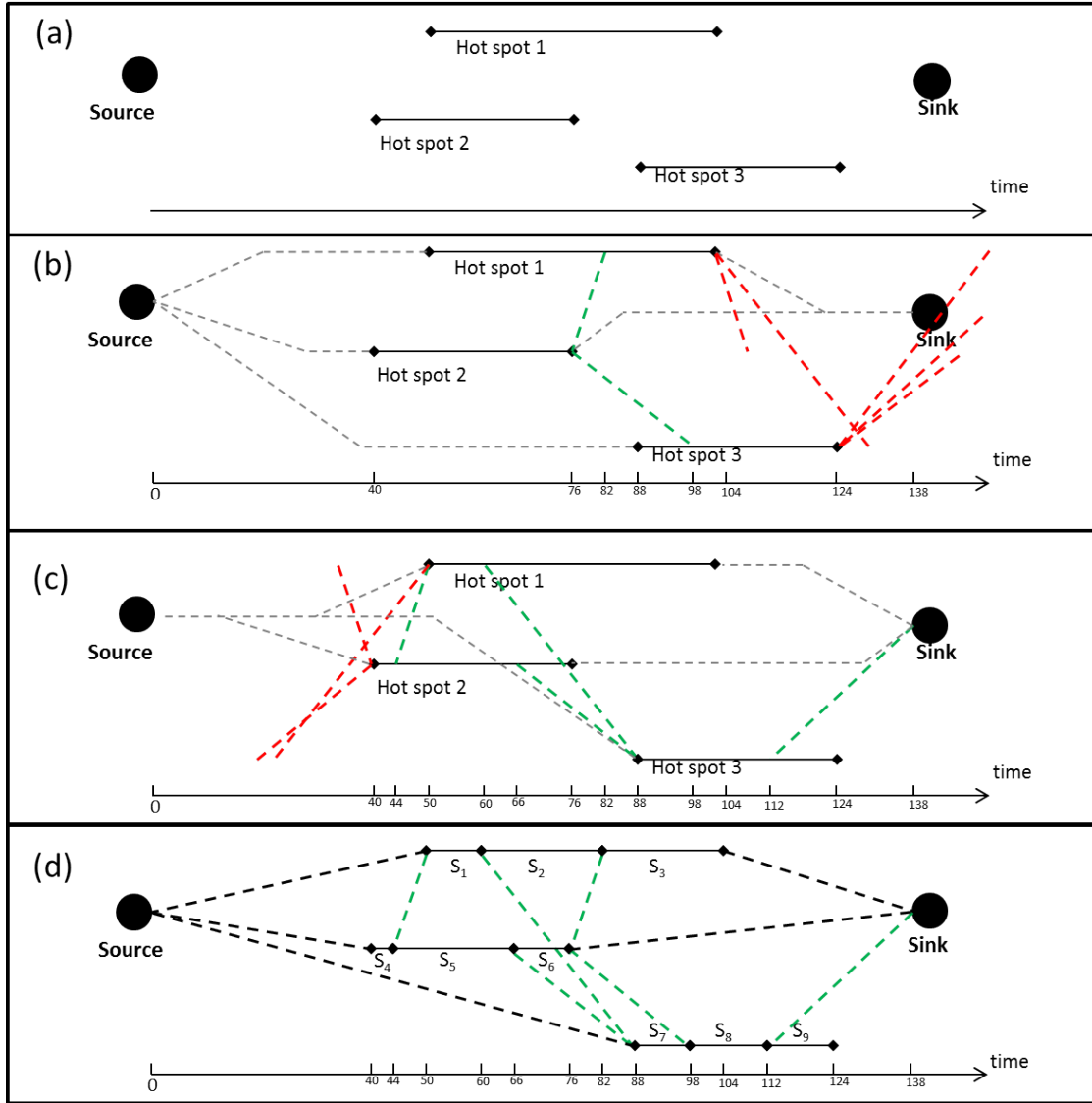


Figure 2: Identifying the *time sections*: (a) a simple network. (b) forward pass. (c) backward pass. (d) sections defined.

time and thus the start or end of a section. The arcs represent traveling from one section to another section or represent the actual patrolling of a section.

$f_{ij}$  represents the number of patrol cars going from node  $i$  to node  $j$ .

$a_{ij}$  represents the gain to be had by traveling from node  $i$  to node  $j$ . For all *travel* arcs this gain equals zero and for the *section* arcs, this gain equals the profit to be made at that section.

$c_{ij}$  represents the capacity of the arc and equals 1 for all

arcs, except for the arc going from the source to the sink. The capacity for most arcs is set to 1 because only a single patrol car is allowed to enter a given section and thus collect the profit from that hot spot for the section's duration of time. The arc going from the source to the sink does not have a capacity constraint and represents cars that do not pass through the physical network but remain at the depot. Since the arcs do not represent physical links, the capacity is unrelated to the actual traffic capacity of a road connection.

$D$  equals the number of patrol cars in the network originating from a single source depot and ending in a single

sink depot. Obviously, the source depot can equal the sink depot.

Consider again the example from figure 2 where nine sections were identified. In figure 3, it is shown that for each section an arc with an *in-node* and an *out-node* is defined. Each such arc has a capacity of 1 and has an  $a_{ij}$  equal to minus the duration multiplied by the "gain per minute" for that section. Since the MCNFP minimizes total cost and the gains per section visit are negative, the total gain will be maximized. Section 9 is not included since a patrol car visiting this time section can never get back to the depot on time. It is possible to retain section 9 in the model. The optimum will still be found. However, since it is easy to recognize that section 9 can not be part of a feasible solution during the section identification phase, it is equally easy to disregard it and, as a consequence, reduce the problem size.

Next, the travel arcs going into the in-nodes and out of the out-nodes are defined (depicted in figure 4). The travel arcs represent only feasible moves i.e. the arc going from the *out-node*  $i$  of section  $I$  to the *in-node*  $j$  of section  $J$  can only be added if the closing time of section  $I$  plus the travel time of section  $I$  to section  $J$  is smaller than or equal to the opening time of the time window at section  $J$ . These arcs going into the *in-nodes* and out of the *out-nodes* have a capacity of one and a cost of zero. Again, it should be noted that it is still physically possible for two cars to be present in the same hot spot at the same time, but this is modeled by having one of the cars assigned to a travel arc that passes by the hot spot location on the physical network. For example, a (non-optimal) feasible solution with two patrol cars to the network in figure 4 could be one car traveling from the source to  $S_1$  and then to the sink. And another traveling from the source to  $S_4$ , then to  $S_2$ , and lastly to the sink. In this case, where  $S_1$  and  $S_2$  are physically in the same location, the arc going from the out-node of  $S_4$  to the in-node of  $S_2$  (shown in red) has to be interpreted by the trooper or dispatcher as either "waiting at the end of  $S_4$ ", "traveling slower" or "arriving at the start of  $S_1$ ", but not incurring the profit of  $S_1$ ". The profit of  $S_1$  is collected by the first car. No arcs that represent a move backward in time are added. Lastly, an uncapacitated arc connecting the source and the sink is added to allow for solutions where one or more cars remain at the depot.

Note that for a problem with  $n$  hot spots, one source, and one sink, each hot spot's time window will at maximum be split into  $n + 2$  sections. Therefore, modeling

an MCPRP with  $n$  hot spots results in an MCNFP with a maximum of  $n * (n + 2)$  sections. In the above example, we end up with considerably less, since many of the combinations result in "red" or "gray" dotted lines which do not create additional sections.

Since the MCNFP is solvable in polynomial time and the transformation of the MCPRP into the MCNFP entails only a polynomial increase in the number of nodes, the MCPRP is solvable in polynomial time. This means that since Keskin et al. [12] prove that their MIP model for the MCPRP is NP-hard, they have unfortunately chosen to utilize an NP-hard model to solve what we show to be an "easy" problem, belonging to P.

#### 2.4. MCPRP MIP formulation of Keskin et al. [12]

The model as presented by Keskin et al. [12] in some cases fails to find feasible solutions because they make use of a big  $M$  variable which actually is too small. Their MIP model is presented in algebraic equations (6)-(22) where  $s_{ik}$  and  $f_{ik}$  represent respectively the start and finish time of a service of a hot spot  $i$  by a car  $k$ .  $t_{ij}$  equals the travel time of hot spot  $i$  to hot spot  $j$  and the binary variable  $x_{ijk}$  equals 1 if car  $k$  travels from hot spot  $j$  to hot spot  $i$  and zero otherwise.  $s_i$  and  $l_i$  equal respectively the earliest service time possible of hot spot  $i$  and the latest service time possible of hot spot  $i$ . Keskin et al. define the big  $M$  variable  $M_{ij}$  as  $M_{ij} = \max(l_i + t_{ij} - e_j, 0)$ . Additional decision variables  $y_{ik}$ ,  $u_{ikg}$  are introduced to avoid double counting the profit when multiple cars visit the same hot spot at the same time.

$$\text{Max} \sum_{i \in N} \sum_{k \in K} (f_{ik} - s_{ik}) \quad (6)$$

$$f_{ik} + t_{ij} - s_{jk} \leq (1 - x_{ijk})M_{ij} \quad \forall k \in K; \forall (i, j) \in \epsilon \quad (7)$$

$$e_i \sum_{j \in \Delta^+(i)} x_{ijk} \leq s_{ik} \quad \forall k \in K; \forall i \in V \quad (8)$$

$$l_i \sum_{j \in \Delta^+(i)} x_{ijk} \geq f_{ik} \quad \forall k \in K; \forall i \in V \quad (9)$$

$$s_{ik} \leq f_{ik} \quad \forall k \in K; \forall i \in V \quad (10)$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1 \quad \forall k \in K \quad (11)$$

$$\sum_{i \in \Delta^-(j)} x_{ijk} = \sum_{i \in \Delta^+(j)} x_{jik} \quad \forall k \in K; \forall j \in N \quad (12)$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \quad \forall k \in K \quad (13)$$

$$\sum_{j \in \Delta^+(i)} x_{ijk} = y_{ik} \quad \forall k \in K; \forall i \in N \quad (14)$$

$$y_{0,k} = y_{n+1,k} = 1 \quad \forall k \in K \quad (15)$$

$$u_{ikg} + u_{igk} \leq y_{ik} \quad \forall i \in V; \forall k, g \in K, g > k \quad (16)$$

$$u_{ikg} + u_{igk} \leq y_{ig} \quad \forall i \in V; \forall k, g \in K, g > k \quad (17)$$

$$u_{ikg} + u_{igk} \geq y_{ik} + y_{ig} - 1 \quad \forall i \in V; \forall k, g \in K, g > k \quad (18)$$

$$f_{ik} - s_{ig} - M(1 - u_{ikg}) \leq 0 \quad \forall i \in V; \forall k, g \in K, g > k \quad (19)$$

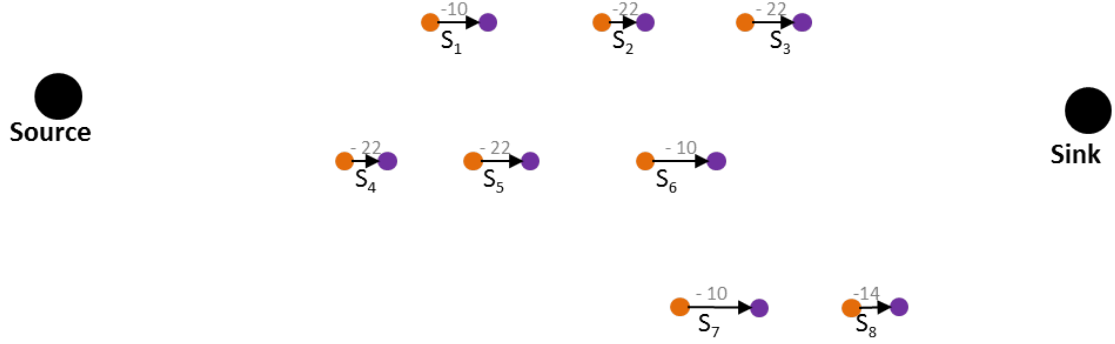


Figure 3: Define an in-and an out-node for each time section and remove time sections that will never be visited.

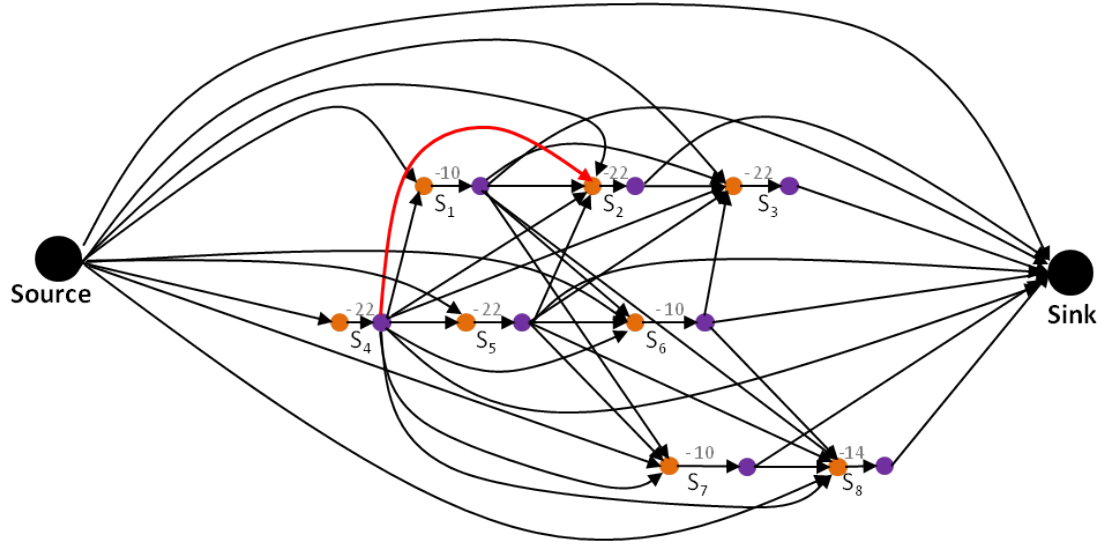


Figure 4: Only add arcs to the network that represent realistic moves, i.e. do not add arcs that represent moving backwards in time.

$$f_{ig} - s_{ik} - M(1 - u_{igk}) \leq 0 \quad \forall i \in V; \forall k, g \in K, g > k \quad (20)$$

$$x_{ijk}, y_{ik}, u_{igk} \in \{0, 1\} \quad \forall i \in V; \forall k, g \in K, g > k \quad (21)$$

$$s_{ik}, f_{ik} \geq 0 \quad \forall i \in V; \forall k \in K, \quad (22)$$

This model is too restrictive. Indeed, consider a solution where hot spot  $H$  is not visited. Then  $\sum_{j \in \Delta^+(H)} x_{Hjk}$  equals zero for all  $k$ , which results in  $s_{Hk} \geq 0$  and  $f_{Hk} \leq 0$ . This, together with (10), means that  $s_{Hk}$  and  $f_{Hk}$  equal zero for all  $k$ . If these two results are entered in (7), for  $j$  equals  $H$ , we obtain:

$$f_{ik} + t_{iH} - 0 \leq (1 - 0)M_{ij} \quad (23)$$

$$f_{ik} + t_{iH} - 0 \leq l_i + t_{iH} - e_H \quad (24)$$

$$f_{ik} \leq l_i - e_H \quad (25)$$

It implies that the latest possible time any arbitrary hot spot  $i$  can be serviced is dependent on the earliest start time of any hot spot  $H$  which is not visited in the solution. This problem can be solved by setting  $M_{ij} = l_i + t_{ij}$ .

That the *profit per minute* is considered the same per

hot spot is another drawback of the mathematical model of Keskin et al. Therefore, they do not differentiate between the accident and/or citation severity of hot spots. Keskin et al. [12] do acknowledge the importance of assigning different weights to hot spots since they perform a set of experiments using a local search heuristic while considering different weights between hot spots. However, the above mathematical model cannot easily be modified to include different gains per minute. Since, if different gains per minute are included, the model has to allow a patrol car to revisit a given hot spot. Revisits have to be allowed, since in some cases where different gains per minute are present, the optimal solution requires revisiting a hot spot. Consider figure 5 where there is one patrol car and two hot spots. Suppose, without loss of generality, that all distances between the hot spots equal zero, hot spot  $A$  is active from time zero un-

til time 480 with a gain per minute of 1, and hot spot  $B$  is active from time 200 until time 220 with a gain per minute of 10. Clearly the optimal solution would be to move to hot spot  $A$  from time 0 until time 200, then move to hot spot  $B$  from time 200 until time 220 and then move back to hot spot  $A$  from time 220 until time 480 resulting in a total gain of 660 minutes. But this is not allowed by the mathematical model of Keskin et al. because a single car can visit a hot spot only a single time since for each combination of hot spot and car, only a single  $s_{ik}$  and  $f_{ik}$  are defined. This results in their model producing an "optimal" solution of only 480 minutes. In contrast, for the MCNFP formulation, this poses no additional difficulty as it only entails changing the  $a_{ij}$  values for the relevant hot spots. The domination rule that allows for the transformation of the MCPRP into the MCNFP remains valid for different gains per minute for the different hot spots. Thus, the MCNFP will return an optimal solution of 660 minutes for the example in figure 5.

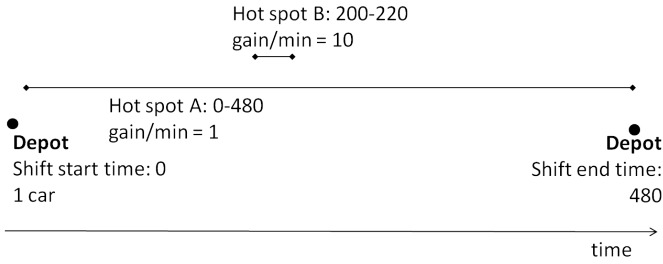


Figure 5: If different *gains per minute* are used, allowing a hot spot to be visited multiple times is a necessity to obtain the optimal solution.

### 3. Computational experiments for the MCPRP instances

The instances of Keskin et al. [12] consist of 60 random instances of 10, 20 and 40 hot spots and 27 real-life instances of 6 to 27 hot spots. All instances were tested with 3 to 8 cars. For the randomized instances, we are not able to compare the objective function values directly with Keskin et al. since in their paper they presented their results relative to the optimal or best solution found for their formulation by Cplex. The Local Search/Tabu Search (LS/TS) heuristic of Keskin et al. reaches the optimal or best solution found by Cplex in 96,7%, 61,7% and 20,8% of the 10, 20 and 40 hot spot instances respectively. The LS/TS heuristic requires up to 15 seconds of computation time for the largest in-

stances on a Dell Poweredge 6850 with four dual-core 3.66 GHz Xeon processors and 8 GB of memory.

Table 1 summarizes the average solution times to reach the optimal solution for the randomized instances with 3 to 8 cars using the MCNFP formulation on an Intel Core Duo E8300 machine with a single dual-core 2.83 GHz processor and 4 GB of RAM. The minimum cost network flow problems are solved using the network simplex algorithm implementation of the LEMON C++ library [13]. It clearly shows that the MCNFP approach including the section identification preprocessing is at least two orders of magnitude faster than the local search/tabu search approach, while reaching the optimal solution for all instances.

Table 1: Average solution times for the MCNFP formulation for the randomized instances of Keskin et al. in seconds.

HS	Number of cars					
	3	4	5	6	7	8
10	0.001	0.001	0.001	0.001	0.001	0.001
20	0.010	0.013	0.009	0.008	0.008	0.008
40	0.107	0.106	0.118	0.116	0.115	0.103

Table 2 shows the comparison of the Local Search/Tabu Search heuristic of Keskin et al. with the optimal MCNFP approach for the real-life instances of Keskin et al. Gain refers to the total amount of minutes the patrol cars are present at a hot spot,  $TW(\%)$  is the total gain divided by the sum of the time windows,  $t_{travel}$  equals the total travel time of the solution,  $t_{dead}$  equals the total time that patrol cars are at a hot spot when the time window is still closed and  $t_{comp}$  equals the computation time required to solve the MCNFP using the network simplex algorithm of the LEMON C++ library, including the preprocessing time. The cars column shows the number of cars available for the instance. If the number is followed by a plus sign, it signifies that our experiments using the MCNFP formulation showed that adding more cars does not increase the objective function. It only results in adding more dead time. From the table we can conclude that, to the merit of Keskin et al., their LS/TS heuristic is able to find the optimal solution in 23 of the 27 instances. In the four instances where the LS/TS approach was not able to find the optimal solution, the average gap equals 4.5%. Çapar et al.[3] mention to be able to also solve all instances to optimality using their improved MIP formulation. A total time of 5163 seconds is required to solve all instances on a powerful 3.6 GHz Intel Xeon Processor with 64 GB RAM. In comparison, the MCNFP approach only requires a total time of 14.54 seconds to solve all instances to optimality. In

addition, the approach of Çapar et al.[3] requires all hot spots to have the same priority, i.e., no differentiation between hot spot severity is possible.

#### 4. Multi-commodity minimum cost flow problem

Although the MCNFP modeling is able to generate many useful insights for policy decision makers such as the number of required patrol cars or the most important hot spots, several other practical extensions cannot be modeled using the MCNFP model. An important extension to the patrol routing problem considers the case where not all patrol cars share the same start and end location and/or do not share the same shift start and end times. The reason for introducing overlapping shifts for patrol cars is that when you have, for instance, just three non-overlapping shifts during a day, around shift change times, all patrol cars are either returning to the depot or on their way for their first hot spot, resulting in no hot spots being covered. This could be solved by overlapping shifts.

Another practical extension involves the introduction of multiple depots. These can be actual depots, temporary stations, as considered by Li and Keskin [14], or even the trooper's homes if they are allowed to start or end their shift from their homes. Moonen et al. [17] consider a multi-depot case where patrol units are allocated to zones (depots) but are allowed to respond to calls in neighboring zones. Such practical problems can most likely be solved more efficiently by modeling certain sub problems as a multi-depot MCNFP.

Simply setting multiple sources and sinks and solving it as an MCNFP could result in infeasible solutions where a specific patrol car does not end up in its designated sink. Ending up in the wrong sink can correspond to ending up in a different end location or can correspond to finishing the shift too early or too late. However, these extensions can be modeled by adopting the multi-commodity minimum cost network flow problem (MCMCNFP). The MCMCNFP aims to route a set of distinct commodity flows through a network. The MCMCNFP is proven to be NP-hard [9]. Nevertheless, extensive research has been conducted and good algorithms are available [11, 16, 1].

The MCMCNFP can be modeled by the following mixed integer program where  $f_{ij}^k$  represents the flow of commodity  $k$  from node  $i$  to node  $j$ ,  $a_{ij}^k$  represents the

cost of moving one unit of commodity  $k$  flow over arc  $ij$ ,  $c_{ij}$  represents the capacity of arc  $ij$  and  $D^k$  represents the required flow of commodity  $k$  with the origin  $o_k$  and destination  $d_k$ .

$$\text{Min} \sum_k \sum_i \sum_j a_{ij}^k f_{ij}^k \quad (26)$$

$$\sum_k f_{ij}^k \leq c_{ij} \quad \forall i, j \quad (27)$$

$$\sum_j f_{ji}^k = \sum_j f_{ij}^k \quad \forall k, \forall i \neq o, d \quad (28)$$

$$\sum_i f_{o_k, i}^k = \sum_i f_{i, d_k}^k = D_k \quad \forall k \quad (29)$$

$$f_{ij}^k \in \{0, 1\} \quad \forall i, j, k \quad (30)$$

Constraint (27) ensures that the flow over an arc never exceeds the arc's capacity, (28) ensures that all flow of a given commodity entering a node, also exits that node and constraint (29) ensures that the commodity flows leave and enter the correct origin and destination locations. And lastly, since the MCMCNFP does not exhibit the integrality property, constraint (30) guarantees binary flow.

The MCPRP with overlapping shifts and different start/end locations for the patrol cars can be modeled as a multi commodity network flow problem by splitting the hot spot time windows in sections as in the single commodity version above. The multi-commodity aspect appears by assigning each unique start/end location and shift combination to a separate commodity  $k$ . Consider a network with 4 cars. Car 1 is stationed at depot  $X$  and is subject to shift schedule  $A$ . Car 2 is stationed at depot  $Y$  and is subject to shift schedule  $B$ . Cars 3 and 4 are also stationed at depot  $Y$  and are subject to shift schedule  $C$ . In this case, there are three distinct commodities ( $k=3$ ) and  $D_1$  and  $D_2$  equal one and  $D_3$  equals 2.

#### 5. Computational experiments for the MCMCNFP instances

The MIP presented above ((26)-(30)) was implemented in Cplex version 12.6 and tested on instances of practical size to test whether solution times remain tractable for the practitioner. The objective of these experiments is not to determine whether a particular instance is served better by utilizing multiple depots or adopting an



Table 2: Comparison of MCNFP results with results of Keskin et al.

Instance	Cars	HS	LS/TS		MCNFP				
			Gain	TW(%)	$t_{travel}$	$t_{dead}$	Gain	TW(%)	$t_{comp}$
Jef MM	8	21	697	<b>86%</b>	736	2407	697	<b>86%</b>	0.015
Jef MA	8	27	933	84%	1132	1738	970	<b>87%</b>	0.040
Jef ME	3+	6	113	<b>63%</b>	237	1091	113	<b>63%</b>	0.001
Jef FM	5+	17	777	<b>81%</b>	674	950	777	<b>81%</b>	0.005
Jef FA	8	19	1213	86%	857	1753	1230	<b>87%</b>	0.018
Jef FE	3	8	286	<b>96%</b>	365	789	286	<b>96%</b>	0.000
Jef SM	5+	18	525	<b>88%</b>	679	1195	525	<b>88%</b>	0.007
Jef SA	4+	14	397	<b>88%</b>	520	1002	397	<b>88%</b>	0.001
Jef SE	5+	16	505	<b>89%</b>	729	1166	505	<b>89%</b>	0.004
Mob MM	5+	20	804	<b>96%</b>	409	1187	804	<b>96%</b>	0.006
Mob MA	6+	17	827	<b>95%</b>	420	1633	827	<b>95%</b>	0.009
Mob ME	4	9	306	<b>93%</b>	166	1448	306	<b>93%</b>	0.001
Mob FM	6	15	870	<b>94%</b>	316	1693	870	<b>94%</b>	0.006
Mob FA	5+	15	1017	<b>97%</b>	268	1116	1017	<b>97%</b>	0.007
Mob FE	5+	10	374	<b>89%</b>	190	1836	374	<b>89%</b>	0.002
Mob SM	6+	19	689	85%	405	1700	775	<b>96%</b>	0.009
Mob SA	5+	21	964	<b>95%</b>	450	986	964	<b>95%</b>	0.010
Mob SE	4+	8	288	<b>96%</b>	204	1429	288	<b>96%</b>	0.001
Tus MM	4+	15	441	<b>92%</b>	335	1144	441	<b>92%</b>	0.003
Tus MA	7	22	854	<b>98%</b>	658	1848	854	<b>98%</b>	0.013
Tus ME	4+	8	241	<b>89%</b>	131	1548	241	<b>89%</b>	0.001
Tus FM	4+	15	534	<b>89%</b>	441	945	534	<b>89%</b>	0.002
Tus FA	5+	15	851	<b>95%</b>	509	1040	851	<b>95%</b>	0.004
Tus FE	4+	8	293	<b>89%</b>	395	1233	293	<b>89%</b>	0.001
Tus SM	5+	9	252	<b>93%</b>	228	1920	252	<b>93%</b>	0.002
Tus SA	3+	13	451	<b>94%</b>	265	724	451	<b>94%</b>	0.001
Tus SE	3	16	410	76%	306	708	426	<b>79%</b>	0.006

overlapping shift schedule. The objective of these experiments is solely to evaluate whether the MCMCNFP can be solved to optimality using off-the-shelf commercial software for instances of practical size.

The multi-commodity instances were adapted from the single commodity instances of Keskin et al. by adjusting each car's start/end location and shift duration. In the original instances, each car's start/end location is location 0 and each car's shift lasts from minute 0 to minute 480. The multi-commodity instances are adapted as follows. Each instance contains two start/end locations: the original depot ( $D1$ ) and a new depot location ( $D2$ ) which is determined by ordering all locations from nearest to furthest from the original depot location and selecting the middle one (if a tie, the furthest is chosen). The shifts are adjusted as summarized in Table 3. As can be seen from the table, the patrol cars are assigned shifts that start an hour earlier, on the same time, half an hour later, or an hour later than the original shifts. In the case of five cars for example, car 1 is stationed at the original depot and its shift lasts from minute -60 until minute 420; car 2 is stationed at the new depot and its shift is the same as car 1; car 3 is

Table 3: Multi-commodity adaptation of instances of Keskin et al.

cars	1	2	3	4	5	6	7	8
	D1	D2	D1	D2	D1	D2	D1	D2
3	-60-420	0-480	60-540					
4	-60-420	-60-420	0-480	60-540				
5	-60-420	-60-420	0-480	0-480	60-540			
6	-60-420	-60-420	0-480	0-480	60-540	60-540		
7	-60-420	-60-420	0-480	0-480	30-510	30-510	60-540	
8	-60-420	-60-420	0-480	0-480	30-510	30-510	60-540	60-540

stationed at the original depot and its shift goes from minute 0 until minute 480, etc.

Tables 4,5,6 and 7 show the solution times and objective function values for all hot spot and car combinations using Cplex version 12.6. The maximum solution times that Cplex required was 660.45 seconds for one of the 20 instances with 40 hot spot and 7 cars and 26.80 seconds for one of the real-life instances. We conclude from this table that the MCMCNFP solution times for these instance sizes, although considerably higher than the MCNFP solution times, remain tractable while offering much greater flexibility in supporting policy decisions. Moreover, these computation times are in the same order of magnitude as the computation times re-

quired by the improved MIP model of Çapar et al.[3]. However, our approach is executed on a much less powerful computer system.

In order to test the computational boundaries of this approach two additional test cases were constructed consisting of 100 and 500 hot spots. It was impossible to run the 500 hot spot instance, even for 3 patrol cars. The number of resulting equations outstripped our computer's memory resources. The 100 hot spot instance was solved to optimality for 3 up to 23 patrol cars. For 24 patrol cars and more, our computer's resources were insufficient. However, if such larger instances need to be considered, our contribution in modeling the original MCPRP as an MCNFP can probably be exploited as a very effective and efficient sub routine or one can turn to specialized MCMCNFP solution methods.

## 6. Discussion and conclusions

This paper modeled the maximum covering and patrol routing problem as presented by Keskin et al. [12] as a minimum cost network flow problem by adding at most a quadratic amount of nodes. This proves that the MCPRP belongs to P and is not NP-hard as was incorrectly concluded by Keskin et al. They unfortunately chose an NP-hard model to represent MCPRP. This underscores the importance of selecting an appropriate model when solving an optimization problem. Computational experiments showed that using a network simplex algorithm, the MCNFP formulation can solve all instances to optimality at least two orders of magnitude faster than the local search/tabu search heuristic proposed by Keskin et al. For the current benchmark instances that are pure MCPRP problems, this does not translate to huge time savings. However, if larger instances are constructed or the MCPRP is a sub problem to a more complex problem, the fact that the MCPRP can be solved by a polynomial algorithm is of major importance.

Practical extensions such as overlapping shifts and different start and end locations of patrol cars can be handled by considering the multi-commodity minimum cost network flow problem. Although the MCMCNFP is proven to be NP-hard, good solution approaches are plentiful in literature. Furthermore, considering the problem sizes of the real-life patrol routing instances, even a straightforward MIP implementation in Cplex can reach the optimal solutions in reasonable compu-

tation times. If larger instances of the multi commodity version have to be solved or other practical extensions such as time-dependent travel times are considered, our contribution of modeling the original MCPRP as an MCNFP can be used to develop an efficient sub-routine for a dedicated algorithm.

A requirement for the transformation of the MCPRP into the MCNFP is that travel times between hot spots are constant. In practice, this is not always the case. For such practical instances, future research might focus on heuristically splitting the hot spots and assuming constant travel times between the newly created hot spots. Since the sub problems can be solved efficiently, this can be embedded in a larger metaheuristic framework or an exact decomposition approach.

In conclusion, the main contributions of this paper are modeling the basic MCPRP as a minimum cost network flow problem, thus proving that the basic MCPRP is not NP-hard and that all available benchmark instances can be solved to optimality in at most a few minutes. We also show that practical extensions to the basic MCPRP can be modeled by a multi-commodity minimum cost network flow problem.

## References

- [1] Aloise, D., Ribeiro, C. C., Mar. 2010. Adaptive memory in multistart heuristics for multicommodity network design. *Journal of Heuristics* 17 (2), 153–179.
- [2] Anderson, T., Jan. 2007. Comparison of spatial methods for measuring road accident hotspots: a case study of London. *Journal of Maps* 3 (1), 55–63.
- [3] Çapar, I., Keskin, B. B., Rubin, P. A., 2015. An improved formulation for the maximum coverage patrol routing problem. *Computers & OR*.
- [4] Chen, X., 2012. Fast Patrol Route Planning in Dynamic Environments. *IEEE Transactions Systems, Man and Cybernetics-Part A: Systems and Humans* 42 (4), 894–904.
- [5] Chircop, P. A., Surendonk, T. J., van den Briel, M., Walsh, T., 2013. A column generation approach for the scheduling of patrol boats to provide complete patrol coverage. In: *20th International Congress on Modelling and Simulation*. No. December. pp. 1–6.
- [6] Divsalar, A. ., Vansteenwegen, P., Sörensen, K., Cattrysse, D. ., 2013. A Memetic Algorithm for the Orienteering Problem with Hotel Selection. *European Journal of Operational Research*, 1–19.
- [7] Divsalar, A., Vansteenwegen, P., Cattrysse, D., Sep. 2013. A variable neighborhood search method for the orienteering problem with hotel selection. *International Journal of Production Economics* 145 (1), 150–160.
- [8] Eiselt, H. A., Gendreau, M., Laporte, G., Laport, G., 1995. Arc Routing Problems, Part 2: The Rural Postman Problem. *Operations Research* 43 (3), 399–414.
- [9] Even, S., Itai, A., Shamir, A., 1976. On the complexity of

Table 4: Computation times (in seconds) and optimal gain values for the artificial MCMCNFP instances with 10 hot spots

cars	3		4		5		6		7		8	
Inst.	t	gain	t	gain	t	gain	t	gain	t	gain	t	gain
1	0.12	256	0.14	256	0.15	258	0.16	258	0.12	258	0.21	258
4	0.13	292	0.11	308	0.17	334	0.14	334	0.20	334	0.33	334
7	0.12	205	0.10	222	0.09	227	0.10	227	0.10	227	0.11	227
10	0.08	212	0.17	212	0.12	215	0.10	215	0.09	215	0.10	215
13	0.08	239	0.11	258	0.09	264	0.25	264	0.09	264	0.28	264
16	0.07	361	0.13	387	0.17	387	0.19	387	0.19	387	0.12	387
19	0.08	251	0.20	251	0.09	251	0.26	251	0.20	251	0.36	251
22	0.07	238	0.11	244	0.12	253	0.26	253	0.20	253	0.20	253
25	0.07	343	0.08	366	0.12	366	0.10	366	0.13	366	0.10	366
28	0.07	338	0.09	346	0.11	348	0.09	348	0.14	348	0.17	348
31	0.07	301	0.13	315	0.13	327	0.19	327	0.12	327	0.25	327
34	0.08	220	0.08	220	0.07	222	0.13	222	0.08	222	0.09	222
37	0.07	228	0.11	228	0.08	245	0.08	245	0.07	245	0.24	245
40	0.06	240	0.11	240	0.15	240	0.14	240	0.10	240	0.19	240
43	0.15	335	0.17	360	0.13	378	0.21	378	0.16	378	0.34	378
46	0.12	219	0.10	219	0.10	247	0.12	247	0.15	247	0.13	247
49	0.09	322	0.14	388	0.17	389	0.15	389	0.14	389	0.16	389
52	0.07	295	0.20	314	0.10	346	0.28	346	0.31	346	0.22	346
55	0.09	288	0.13	288	0.16	288	0.10	288	0.15	288	0.13	288
58	0.17	330	0.21	363	0.23	375	0.35	375	0.34	375	0.31	375
avg	0.09		0.13		0.13		0.17		0.15		0.20	
max	0.17		0.21		0.23		0.35		0.34		0.36	

- timetable and multi-commodity flow problems. SIAM Journal on Computing 5 (4), 691–703.
- [10] Glover, F., Klingman, D., Phillips, N., 1990. Netform Modeling and Applications. Interfaces 20 (August 1990), 7–27.
- [11] Kamath, A., 1995. Fast Approximation Algorithm for Minimum Cost Multicommodity Flow. In: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms. pp. 493–501.
- [12] Keskin, B. B., Li, S. R., Steil, D., Spiller, S., Jan. 2012. Analysis of an integrated maximum covering and patrol routing problem. Transportation Research Part E: Logistics and Transportation Review 48 (1), 215–232.
- [13] Király, Z., Kovács, P., 2012. Efficient implementations of minimum-cost flow algorithms. Acta Universitatis Sapientiae, Informatica, 67–118.
- [14] Li, S. R., Keskin, B. B., Nov. 2013. Bi-criteria dynamic location-routing problem for patrol coverage. Journal of the Operational Research Society, 1–15.
- [15] Lou, Y., Yin, Y., Lawphongpanich, S., Apr. 2011. Freeway service patrol deployment planning for incident management and congestion mitigation. Transportation Research Part C: Emerging Technologies 19 (2), 283–295.
- [16] McBride, R. D., 1998. Advances in Solving the Multicommodity-Flow Problem. Interfaces 28 (2), 32–41.
- [17] Moonen, M., Cattrysse, D., Oudheusden, D. V., 2008. Organising Patrol Deployment against Violent Crimes. Operations Research: an International Journal 7 (3), 401–417.
- [18] Orlin, J. B., Aug. 1997. A polynomial time primal network simplex algorithm for minimum cost flows. Mathematical Programming 78 (2), 109–129.
- [19] Portugal, D., Rocha, R. P., Dec. 2013. Distributed multi-robot patrol: A scalable and fault-tolerant framework. Robotics and Autonomous Systems 61 (12), 1572–1587.
- [20] Steil, D., Parrish, A., 2009. HIT : A GIS-Based Hotspot Identification Taxonomy. I. J. Comput. Appl 16 (2), 1–10.
- [21] Takamiya, M., Watanabe, T., 2011. Planning High Responsive Police Patrol Routes with Frequency Constraints. In: Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication.
- [22] Tomlin, J., 1966. Minimum-cost multicommodity network flows. Operations Research 14 (1), 45–52.
- [23] Tsiligrirides, T., 1984. Heuristic Methods Applied to Orienteering. The Journal of the Operational Research Society 35 (9), 797–809.
- [24] Vansteenwegen, P., Souffriau, W., Van Oudheusden, D., Feb. 2011. The orienteering problem: A survey. European Journal of Operational Research 209 (1), 1–10.
- [25] Willemse, E. J., Joubert, J. W., Jun. 2011. Applying minmax k postmen problems to the routing of security guards. Journal of the Operational Research Society 63 (2), 245–260.
- [26] Winston, L. W., 1987. Operations research: Applications and algorithms. Duxbury Press, Boston.

## Appendix A. Proof of lemma 1

This proof considers two hot spots with partly overlapping time windows (figure A.6). The gain per minute equals  $g_1$  and  $g_2$  for respectively hot spot  $I$  and hot spot 2 and the travel time between both hot spots equals  $d_{ij}$ . The time window of hot spot  $I$  opens at time  $t_{o,1}$  and closes at time  $t_{c,1}$ . Similarly, the time window of hot spot  $I$  opens at time  $t_{o,2}$  and closes at time  $t_{c,2}$ .

Lemma 1 states that one of the optimal solutions consists of all cars arriving at hot spots  $i$  either at the open-

ing of their time window or at a time that corresponds to the closing time of another hot spot  $j$  plus the travel time between  $i$  and  $j$ . Similarly, all cars will only leave a hot spot at the closing of their time window or at a time that corresponds to the opening time of another hot spot  $j$  minus the travel time between  $i$  and  $j$ .

When considering two partly overlapping hot spots in isolation, lemma 1 states that leaving hot spot 1 to move to hot spot 2 at any time  $T$  will be dominated by either leaving at  $t_{b,21}$  or  $t_{c,1}$ .

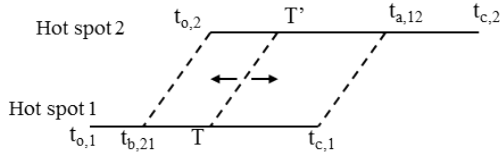


Figure A.6: Partly overlapping time windows.

Leaving hot spot 1 before  $t_{b,21}$  or after  $t_{c,1}$  is sub optimal, since it results in dead waiting time where no gain is collected.

Considering the case where hot spot 1 has a higher gain per minute than hot spot 2, the patrol car in an optimal solution will remain until  $t_{c,1}$ , before moving to hot spot 2. Leaving earlier at time  $T$  results in earning  $g_1 * (t_{c,1} - T)$  less in hot spot 1, but earning  $g_2 * (t_{a,12} - T')$  more in hot spot 2. Since  $T'$  equals  $T + d_{ij}$  and  $t_{a,12}$  equals  $t_{c,1} + d_{ij}$ , leaving earlier results in a change of gain earned of  $(g_2 - g_1)(t_{c,1} - T)$ . Since  $g_1$  is larger than  $g_2$ , this solution is worse than remaining in hot spot 1 until the end of the time window.

Analogously, if hot spot 2 has a higher gain per minute than hot spot 1, the patrol car will leave hot spot 1 as soon as possible. Leaving later will reduce the total gain earned.

If both hot spots have the same gain per minute, leaving hot spot 1 at  $t_{b,21}$  or  $t_{c,1}$  will be as good as leaving at any time  $T$  in between  $t_{b,21}$  and  $t_{c,1}$ .

Since an identical reasoning can be applied to any combination of hot spots in the network, this concludes the proof.

Table 5: Computation times (in seconds) and optimal gain values for the artificial MCMCNFP instances with 20 hot spots

cars	3		4		5		6		7		8	
Inst.	t	gain	t	gain	t	gain	t	gain	t	gain	t	gain
2	0.38	499	0.59	604	1.28	658	2.12	675	2.01	675	2.10	675
5	0.47	614	0.96	692	0.88	760	1.25	782	2.09	783	2.02	783
8	0.31	460	0.92	511	0.92	563	1.47	568	1.24	574	1.65	578
11	0.16	344	0.21	409	0.35	410	0.70	418	0.51	420	0.69	426
14	0.24	485	0.64	557	0.71	576	1.22	578	1.73	578	1.73	578
17	0.27	568	0.46	680	0.76	728	1.01	738	1.59	748	2.73	749
20	0.32	555	0.62	615	0.69	640	1.07	641	0.94	645	1.05	645
23	0.49	467	1.01	532	1.58	564	2.23	569	1.62	569	3.67	569
26	0.27	474	0.55	561	1.07	620	1.49	646	1.78	662	1.56	667
29	0.21	531	0.67	586	0.80	599	1.03	599	1.14	599	1.30	599
32	0.26	439	0.37	484	0.50	480	0.72	495	0.62	495	0.93	495
35	0.20	514	0.52	564	0.62	567	0.69	567	0.80	567	1.54	567
38	0.43	509	1.20	567	1.58	617	2.61	621	2.46	621	3.69	621
41	0.34	470	0.49	547	0.78	598	1.02	619	1.21	636	1.12	636
44	0.31	416	0.29	467	0.77	480	0.83	480	0.89	480	0.89	480
47	0.40	451	0.69	508	0.81	563	1.03	567	0.95	587	1.04	596
50	0.57	476	0.96	552	1.14	574	1.88	599	1.99	622	2.54	624
53	0.30	572	0.70	690	0.76	726	1.04	726	1.53	727	1.51	727
56	0.15	442	0.20	466	0.40	484	0.56	484	0.49	484	0.53	484
59	0.35	483	0.46	588	0.52	595	1.69	599	2.25	599	2.48	599
avg	0.32		0.62		0.85		1.28		1.39		1.74	
max	0.57		1.20		1.58		2.61		2.46		3.69	

Table 6: Computation times (in seconds) and optimal gain values for the artificial MCMCNFP instances with 40 hot spots

cars	3		4		5		6		7		8	
Inst.	t	gain	t	gain	t	gain	t	gain	t	gain	t	gain
3	3.79	737	7.46	937	20.31	1011	32.98	1081	71.30	1104	80.52	1129
6	7.10	691	20.10	891	26.26	1033	67.36	1134	211.68	1215	221.93	1242
9	3.56	627	16.63	802	20.10	890	37.91	954	75.99	991	124.65	1029
12	5.98	651	17.20	850	36.58	997	59.17	1072	71.87	1121	193.64	1149
15	4.29	640	12.82	842	19.23	980	35.96	1112	65.68	1191	103.78	1238
18	4.00	619	7.59	764	12.13	922	27.14	982	40.22	1039	65.70	1066
21	3.94	639	12.28	817	24.48	964	34.12	1066	87.73	1142	107.13	1174
24	4.42	685	10.94	878	25.64	1023	45.40	1114	123.93	1199	137.28	1248
27	4.93	698	17.10	855	19.73	1010	46.57	1087	141.35	1150	250.38	1184
30	4.91	743	17.78	936	26.50	1087	75.94	1194	117.50	1247	173.04	1285
33	4.03	658	14.80	853	22.88	989	46.40	1138	89.36	1216	171.77	1276
36	5.06	690	11.65	864	30.71	969	72.10	1014	102.77	1056	131.16	1078
39	6.80	660	19.97	852	35.25	916	56.71	999	91.13	1067	348.04	1129
42	5.83	749	12.92	950	41.30	1105	58.76	1193	660.45	1232	153.60	1250
45	4.50	661	10.09	852	30.27	1000	52.81	1097	80.05	1163	214.73	1203
48	7.59	657	25.26	800	43.12	934	58.97	977	94.42	1020	162.19	1052
51	4.96	744	15.62	925	27.67	1037	56.37	1099	71.77	1126	90.30	1149
54	4.38	625	9.71	799	23.16	938	29.64	1020	47.69	1120	73.10	1152
57	9.68	635	24.03	852	41.13	948	76.18	1030	119.44	1088	201.10	1127
60	6.27	622	15.82	825	40.43	933	61.34	1035	73.79	1116	196.30	1175
avg	5.30		14.99		28.34		51.59		121.90		160.02	
max	9.68		25.26		43.12		76.18		660.45		348.04	

Table 7: Computation times (in seconds) and optimal gain values for the real-life MCMCNFP instances

cars		3		4		5		6		7		8	
Inst.	t	gain	t	gain	t	gain	t	gain	t	gain	t	gain	
Jef FA	0.68	755	1.57	939	1.68	1101	3.29	1147	5.22	1251	5.08	1311	
Jef FE	0.06	251	0.07	299	0.11	299	0.13	299	0.13	299	0.08	299	
Jef FM	0.15	664	0.32	811	0.42	871	0.49	889	0.68	914	0.71	914	
Jef MA	1.69	666	5.58	771	8.59	891	10.34	929	20.98	982	26.80	1012	
Jef ME	0.05	142	0.07	160	0.09	160	0.09	179	0.11	175	0.17	179	
Jef MM	0.63	478	1.21	561	1.27	624	1.58	629	2.46	686	2.83	707	
Jef SA	0.09	363	0.11	412	0.16	437	0.12	449	0.15	449	0.15	449	
Jef SE	0.15	495	0.24	525	0.40	543	0.38	570	0.48	570	0.69	570	
Jef SM	0.24	457	0.36	531	0.81	559	0.94	589	1.09	588	1.46	590	
Mob FA	0.30	816	0.54	959	0.65	1020	0.71	1020	1.39	1042	1.21	1042	
Mob FE	0.12	293	0.11	352	0.11	401	0.33	412	0.28	420	0.13	420	
Mob FM	0.23	705	0.30	783	0.77	843	1.07	885	1.49	925	1.24	930	
Mob MA	0.60	614	0.97	718	1.47	797	1.43	800	2.24	849	3.22	849	
Mob ME	0.11	278	0.11	324	0.24	330	0.26	330	0.29	330	0.16	330	
Mob MM	0.19	701	0.56	798	0.72	833	0.98	833	1.61	840	2.24	840	
Mob SA	0.54	794	0.82	931	1.21	989	1.43	989	1.27	996	1.60	996	
Mob SE	0.08	265	0.16	294	0.21	299	0.22	299	0.23	299	0.27	299	
Mob SM	0.36	613	0.84	712	1.06	786	3.53	805	2.80	810	2.27	810	
Tus FA	0.16	674	0.33	786	0.47	855	0.53	856	0.55	886	0.96	886	
Tus FE	0.08	307	0.11	330	0.14	330	0.16	330	0.11	330	0.15	330	
Tus FM	0.14	503	0.22	576	0.27	584	0.28	595	0.32	600	0.31	600	
Tus MA	0.73	580	1.13	662	1.66	722	2.75	752	4.05	844	5.44	857	
Tus ME	0.06	245	0.11	259	0.08	270	0.16	270	0.12	270	0.15	270	
Tus MM	0.11	429	0.16	460	0.25	468	0.25	475	0.29	480	0.53	480	
Tus SA	0.13	477	0.17	480	0.15	480	0.21	480	0.12	480	0.23	480	
Tus SE	0.32	410	0.50	463	0.66	514	0.86	531	0.76	539	1.26	539	
Tus SM	0.09	212	0.16	242	0.19	266	0.13	270	0.24	270	0.24	270	
avg	0.30		0.62		0.88		1.21		1.83		2.21		
max	1.69		5.58		8.59		10.34		20.98		26.80		